

# HoGent

BEDRIJF  
EN  
ORGANISATIE

## Adapter oefeningen

# De calculator

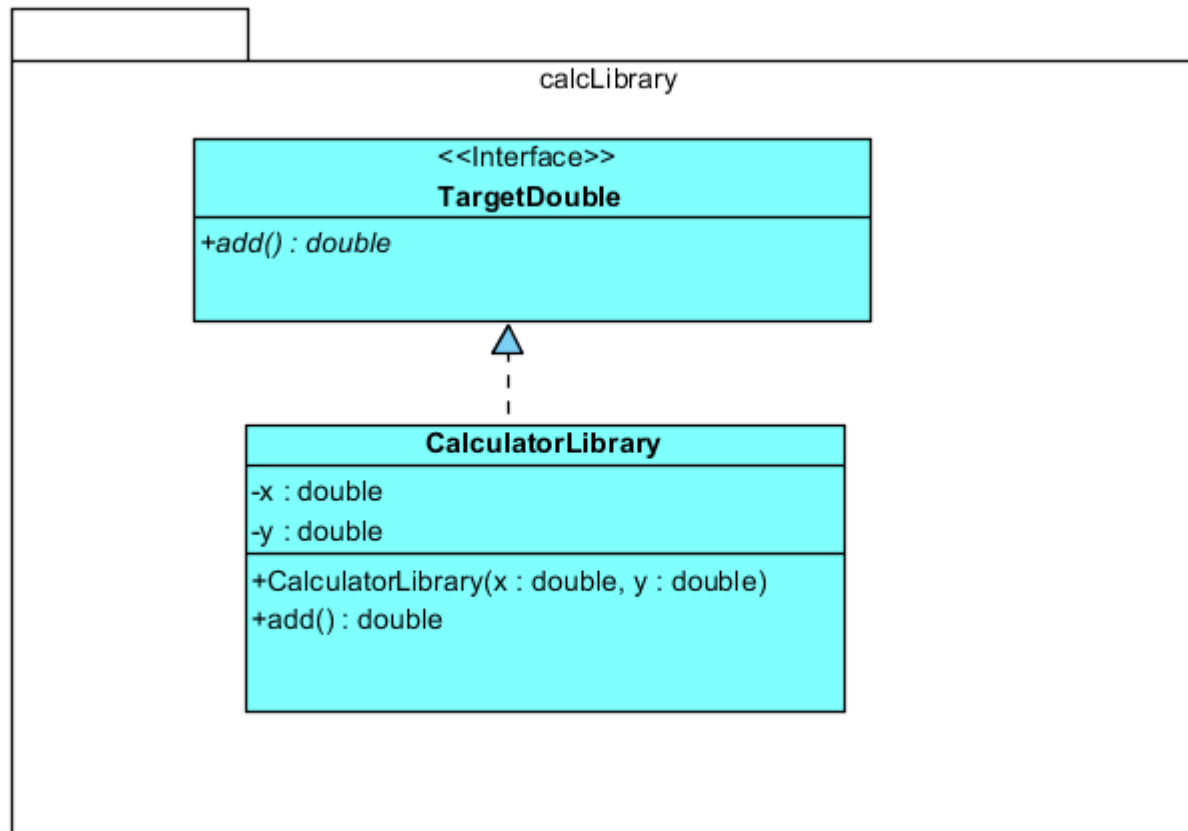
- ▶ De client-code

```
Calculator calculator = new Calculator(2, 3);  
long res = calculator.add();  
System.out.println(res);
```

- ▶ Calculator maakte gebruik van een library die de som van twee gehele getallen berekent. Door een upgrade wijzigt de library. De library werkt nu met decimale getallen.

# De calculator

- ▶ De nieuwe calculator library
  - Voor jar file library : zie de folder ExterneLibraries van deze oefening



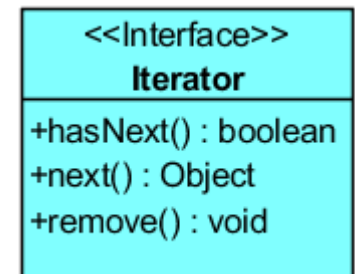
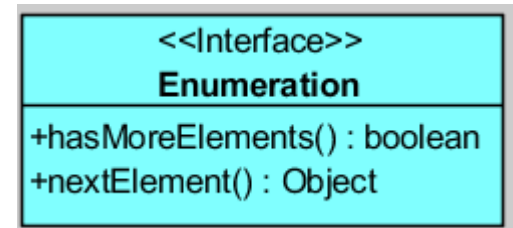
# De calculator

- ▶ Schrijf de code in de Calculator zodat met deze nieuwe library gewerkt wordt. Zoek een oplossing zodat we de client-code niet moeten wijzigen.

```
Calculator calculator = new Calculator(2, 3);  
long res = calculator.add();  
System.out.println(res);
```

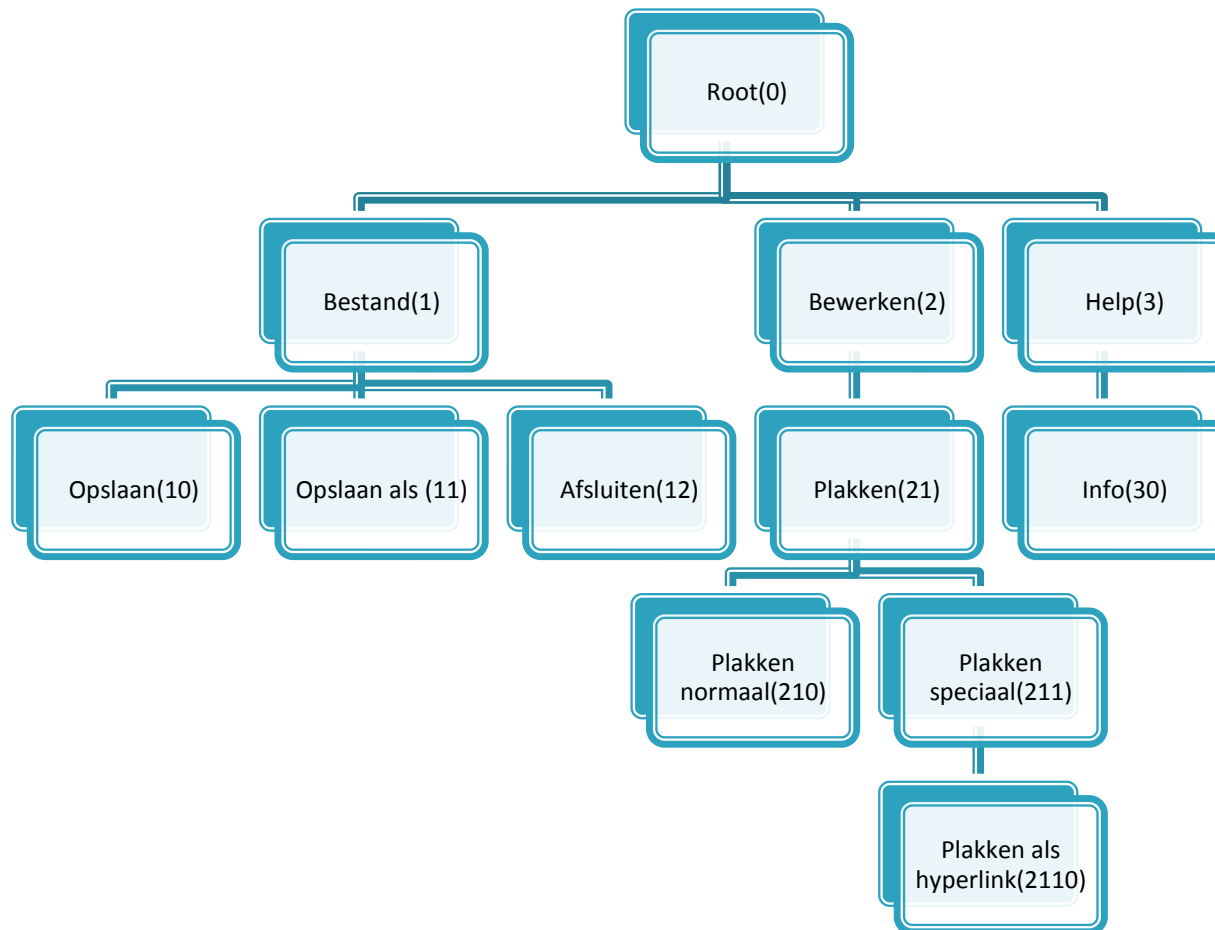
# Adapters in de praktijk

- ▶ Enumerators in de oude wereld (Vector, Stack, Hashtable, StringTokenizer, ...)
  - `hasMoreElements` : vertelt je of er nog elementen in de collectie zitten
  - `nextElement` : levert het volgende element in de collectie op
- ▶ Iterators in de nieuwe wereld
  - `hasNext`: vertelt je of er nog elementen in de collectie zitten
  - `next` : levert het volgende element in de collectie op
- ▶ Pas een Enumeration aan een iterator aan en omgekeerd



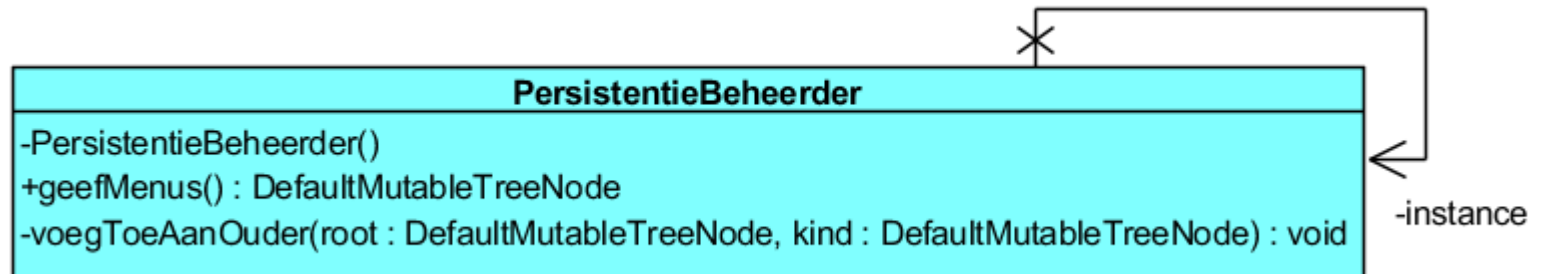
# Deel 1 : Een Enumeration aan een Iterator aanpassen

- ▶ De klasse PersistentieBeheerder
  - De klasse PersistentieBeheerder bouwt een boomstructuur van menu's, submenu's en menu items op



# Deel 1 : Enumeration -> Iterator

## ► De klasse PersistentieBeheerder



- `geefMenus` bouwt de boomstructuur op vertrekkende van een tekstbestand
- `voegToeAanOuder` : voegt een menu/item toe op de juiste plaats in de boomstructuur

# Deel 1 : Enumeration -> Iterator

- ▶ PersistentieBeheerder : geefMenus
  - leest een tekstbestand met de onderstaande layout.

ouderID	IDnr	methode	omschrijving
0	1	<menu>	Bestand
1	10	opslaan	Opslaan
1	11	opslaanAls	Opslaan als
1	12	afsluiten	Afsluiten
0	2	<menu>	Bewerken
2	20	undo	Ongedaan maken
2	21	<menu>	Plakken
21	210	paste	Plakken normaal
21	211	<menu>	Plakken speciaal
211	2110	pasteHyperlink	Plakken als Hyperlink
0	3	<menu>	Help
3	30	info	Info

- Dit bestand bevat de menu's, submenu's en menu items. Om de structuur van het hele menu bij te houden krijgt elk item/menu een IDnr, en een ouderID. methode bevat de bijhorende methodenaam die geactiveerd moeten worden als een item wordt gekozen.



# Deel 1 : Enumeration -> Iterator

- ▶ PersistentieBeheerder : geefMenus.
  - Van elke regel van de tekstfile wordt een object gemaakt van de klasse **MenuActie**, met vier attributen **ouderId**, **menuActieId**, **omschrijving** en **actie**
  - Deze MenuActie objecten plaatsen we in een **boomstructuur** waarbij de root (id 0) enkel menu's als kinderen krijgt. De menu's zelf krijgen als kinderen de menu-items en/of submenu's. Submenu's zijn eigenlijk ook gewoon menu's die zelf ook weer kinderen kunnen hebben.
  - De volgorde van de tekstregels in het tekstbestand is niet van belang, zolang ouderknopen voor kinderknopen geleverd worden.

# Deel 1 : Enumeration -> Iterator

- ▶ PersistentieBeheerder : geefMenus
  - Retourneert een DefaultMutableTreeNode
    - Java klasse voor willekeurige boomstructuren
    - Is niet generiek: datadeel wordt bijgehouden als object (get/set UserObject).
    - In de oefening wordt als data in de node een MenuActie gebruikt.
    - Reeds aanwezig voor de klasse Iterator in java bestond, werkt daarom nog met Enumeration.

# Deel 1 : Enumeration -> Iterator

- ▶ PersistentieBeheerder : voegToeAanOuder
  - De methode VoegToeAanOuder voegt een node toe aan de juiste parent. Hiervoor wordt de boomstructuur doorlopen, gebruik makend van een Enumeration.

## depthFirstEnumeration

```
public Enumeration depthFirstEnumeration()
```

Creates and returns an enumeration that traverses the subtree rooted at this node in depth-first order. The first node returned by the enumeration's `nextElement()` method is the leftmost leaf. This is the same as a postorder traversal.

Modifying the tree by inserting, removing, or moving a node invalidates any enumerations created before the modification.

### Returns:

an enumeration for traversing the tree in depth-first order

# Deel 1 : Enumeration -> Iterator

- ▶ Maak gebruik van een iterator voor het doorlopen van de boom.
  - Maak de adapter EnumerationIterator<E>
  - Pas de code aan in de PersistentieBeheerder
  - De PersistentieTest bevat de testen die na de aanpassing moeten slagen.

# Deel 2 : Een Iterator aan een Enumeration aanpassen

## ► De klasse Class1

- De klasse Class1 bevat een lijst van strings. In de ToString methode worden de strings na mekaar geplaatst. Hiervoor wordt gebruik gemaakt van een Iterator.
- Maak gebruik van een enumeration voor het doorlopen van de lijst.
  - Maak de adapter IteratorEnumeration<E>
  - Pas de code aan in de klasse Class1
  - De VanIteratorNaarEnumeratorTest bevat de testen die na de aanpassing moeten slagen.